

Loyola Marymount University
Department of Computer Science

Head Soccer

Jose Fuentes

2D Soccer Repository Link

Supervisor: **Professor Edmiston and Professor Morgan**

December 16, 2023

Contents

1	Introduction	3
2	Project Scope	4
2.1	Starting Point of Project	4
2.2	Project MVP	5
2.3	Stretch Goals	6
3	Project Pitch	7
3.1	Problem	7
3.2	Purpose/Why	7
4	Project Timeline	9
4.1	"Refactoring"	9
4.2	New Features	10
4.3	Art + Final Adjustments	12
5	Testing	13
6	Project Installation Guide	14
7	Passing the Baton	15
7.1	On-boarding	15
7.2	What's left?	16
8	Final Thoughts	18
8.1	What went well?	18
8.2	What went wrong?	19
8.3	What would I have done differently?	19

Introduction

My project is based on two games that were significant parts of my childhood: one called Head Soccer and another called Sports Head Football Championship (SHFC). Both games were 2D soccer games, but they had different visions. I aimed to recreate a game similar to SHFC because, under the hood, it used Adobe Flash, which was discontinued at the end of 2020 along with many other games.

The game is in a 2D format, where the main goal, as you would expect from the title, is to score more goals than the opponent. My game will feature various game modes, including a two-player mode for simultaneous play on the same screen and a single-player mode, resembling more of a campaign or story mode. Similar to real life, the player must participate in numerous games, striving to win as many as possible to accumulate the most points and claim the trophy at the end of the season. Along the way, players can boost their stats, making it easier for them to compete against tougher opponents. During the game, various power-ups (and power-downs, you could say) will be available, aiding either the player or the "enemy" to influence the outcome. Think of the story mode as a typical soccer season, where the objective is to be the best team across the entire season and secure the league trophy.

Project Scope

2.1 Starting Point of Project

As this is a continuing project from last semester during my CMSI 3072 Game Development Class, I would like to explain my starting point.

My project initially comprised only two different "pages": one for the home screen and the actual gameplay. The home screen featured two buttons—one to play and one to quit—surrounded by controls for the game. As you can imagine, the home screen was quite cluttered with lots of information, lacking much room to breathe.

For the gameplay itself, it included the standard players, goals, timer, and score functionality, although some of these features were a bit buggy. For instance, when the score became too large for either player, the game would typically declare the player with the least amount of goals as the winner. Another issue was occasional control stickiness, where the players' controls would get stuck for a few seconds before resuming normal behavior for no apparent reason. Regarding the goals, when the ball entered the goal, it would sometimes get stuck inside for a few seconds before respawning back into the playing field, disrupting the flow of the game.

I also implemented sounds in the game, but there were issues with constant repetition. Sometimes the sounds lingered for too long, while at other times, there was complete silence.

Below are pictures of what the game looked like visually before starting my revamp. The first picture depicts the gameplay, where players engage in the game. The second picture

showcases the home screen, the first thing that players see.

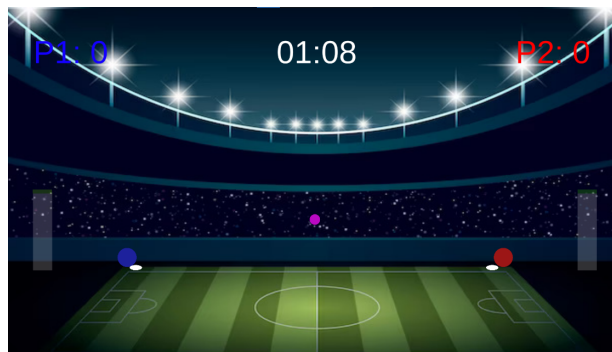


Figure 2.1: Image of the gameplay

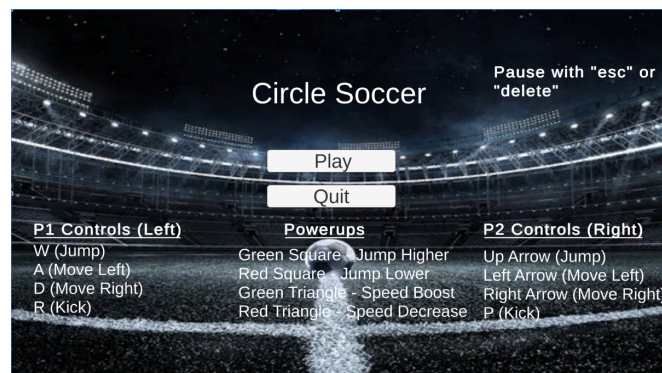


Figure 2.2: Image of the home screen

2.2 Project MVP

Given the issues I discovered in my code around halfway through the semester, my Project MVP shifted from the initial vision I had at the beginning of the semester. The original goals I set became my stretch goals, which will be described shortly. However, the primary objective with the complete rewrite of my code in mind was the following:

1. Ensure basic functionality works as intended from the start: no bugs or issues with goals, player inputs, and score tracking.
2. Once basic functionality is established, focus on refining the home page and creating separate pages for controls and power-ups to enhance the gaming experience and improve visual aesthetics.

3. Develop a form of AI bot capable of at least following the ball and attempting to kick when the ball is nearby. It doesn't need to be the smartest bot at this stage but should serve as a solid foundation for future improvements. This will be the foundation for my single-player mode, excluding the story aspect.

2.3 Stretch Goals

Here was my initial vision for the game at the beginning of the semester, but I came to realize that these goals would become my stretch goals.

1. In the single-player mode, incorporate a narrative aspect where the player faces multiple teams, mirroring a typical soccer season. The objective is to win as many games as possible to accumulate points. Additionally, currency earned after each game allows the player to upgrade their stats, enhancing their chances of winning the season.
2. Introduce a quick-play mode alongside the single-player mode. This mode allows players to choose from three difficulty levels-easy, medium, or hard-indicating the level of the AI bot they will face.
3. Enhance the game's excitement and unpredictability by incorporating power-ups and power-downs. Examples include speed increase/decrease, jump boost/reduction, variations in ball bounce, and changes in goal size.
4. Integrate artwork into the game. Currently, basic shapes represent various elements. For instance, rectangles represent goals, and circles denote players and bots. Consider creating 2D pixel art to fill this gap or finding suitable artwork online that aligns with the game's needs.

Project Pitch

3.1 Problem

It's safe to say that at least once in our lives, we've played on sites like Cool Math Games and enjoyed numerous games that utilized Adobe Flash. As you may be aware (or perhaps not), Flash was discontinued at the end of 2020. This means that many of the games you might have played in the past either no longer exist or are only available on very niche websites that are challenging to locate. One of these games, which I have fond memories of, is called Sports Head Football Championship. I used to play this game frequently as a kid and even introduced it to others. We would organize big tournaments to determine who the better player was. Unfortunately, I can no longer find this game through conventional means (unless, for some reason, it's on the dark web).

3.2 Purpose/Why

The reason I am attempting to recreate this game and add my twist to it is simple: it was a significant part of my childhood and that of many of my close friends. Unlike other Flash games that some have managed to recreate, I noticed that this particular game was overlooked for a revamp. I aim to bring joy to the little kid inside me by producing a game I always used to play as a child.

Another reason for choosing this project is that I *thought* I had an underlying foundation

to work off, given that I started this project back in the Spring 2023 semester for my Game Development class. I briefly discussed this in section 2.1, "Starting Point of Project," but to recap, my code wasn't clean or robust enough to achieve the things and vision I had for this game. This led me to rewrite my entire foundational code from scratch, impacting my progress, as discussed further in the "Timeline" section.

Furthermore, I have a keen interest in the gaming industry and may one day be involved in game projects, creating games for a living. Despite being aware of the industry's stigma compared to other fields in Computer Science-such as concerns about work-life balance, lower pay, and overall challenging working conditions-I believe this shouldn't deter me from pursuing something I genuinely enjoy. In the worse-case scenario, where finding a job or landing a job becomes challenging, I could leverage the skills learned in school and potentially in the workforce as a side hobby to create my games at my own pace. So, why not enhance my skills further and become more of an expert in this subject?

Project Timeline

My project can be divided into approximately three to four sections or milestones throughout the semester: "Refactoring," "New Features," "Art," and "Final Adjustments." While there is an argument for combining "New Features" and "Art" into a single category, I chose to separate them into their own sections. This division is based on considering "New Features" as more of the technical and programmatic aspects, while "Art" is seen as a non-technical aspect.

You can explore the specific components of the game that I have completed on my [Trello Board](#).

4.1 "Refactoring"

I addressed this in Section 2.1 - "Starting Point of Project," where I outlined the major issues and flaws in my game, mistakenly thinking I had a solid base to work with. I won't reiterate those mistakes in this section, but I want to emphasize that refactoring consumed the majority of my time this semester.

Initially, my timeline for refactoring my code was limited to the first two months of the semester, covering Early September to mid-October. My deadline for completing the refactoring was set for the 25th of October, with the plan to then start implementing the new features from the 27th onward. Unfortunately, as depicted in the two images below that contain my original timeline, I had hoped to finish my refactoring by the 25th of October and

then begin implementing the new features on the 27th. However, this didn't come to fruition.

	A	B	C	D	E	F	G	H	I	J
1	Spreadsheet		Today's Date		Sep	Sep	Oct	Oct	Oct	Oct
2	Class		10/17/2023		Wed	Wed	Wed	Wed	Wed	Wed
4	Task	Start Date	End Date	Priority	9/20	9/27	10/4	10/11	10/18	10/25
7	Refactor Code									
8	changed to new input system	9/13/2023	9/29/2023	1- (Normal)	changed to new input system					
9	fix goal code	9/13/2023	9/20/2023		fix goal code					
10	fix kicking goal	9/13/2023	9/27/2023		fix kicking goal					
11	fix post code	9/13/2023	9/27/2023		fix post code					
12	fix score code	9/13/2023	10/25/2023	3- (Critical)	fix score code					
13	game manager	9/13/2023	10/25/2023	3- (Critical)	game manager					
14	fix powerups	9/13/2023	12/15/2023	2- (Important)	fix powerups					
15										
16	Design									
17	create control page	9/20/2023	9/27/2023	1- (Normal)	create control page					
18	create powerup page	9/20/2023	9/27/2023	1- (Normal)	create powerup page					
19	separate two-player and single player mode	9/20/2023	9/27/2023	1- (Normal)	separate two-player and single player mode					

	A	B	C	I	J	K	L	M	N	O	P	Q	R	S
1	Spreadsheet		Today's Date	Oct	Oct	Nov	Nov	Nov	Nov	Nov	Dec	Dec	Dec	Dec
2	Class		10/17/2023	Wed	Wed	Wed	Wed	Wed	Wed	Wed	Wed	Wed	Wed	Wed
4	Task	Start Date	End Date	10/18	10/25	11/1	11/8	11/15	11/22	11/29	12/6	12/13	12/20	12/27
21	Single Player Mode													
22	create main menu	10/27/2023	12/15/2023		create main menu									
23	A.I bot	10/27/2023	12/15/2023		A.I bot									
24	matchup generator	10/27/2023	12/15/2023		matchup generator									
25	table creation + update	10/27/2023	12/15/2023		table creation + update									
26	currency	10/27/2023	12/15/2023		currency									
27	upgrade stats	10/27/2023	12/15/2023		upgrade stats									
28														
29	Powerups	11/15/2023	12/15/2023					Powerups						
30	refactor	11/15/2023	12/15/2023					refactor						
31	freeze	11/15/2023	12/15/2023					freeze						
32	ball smaller	11/15/2023	12/15/2023					ball smaller						
33														
34														

4.2 New Features

Given that I couldn't meet my original deadline due to the excessive amount of work needed to rewrite the code essentially from scratch, I didn't complete the rewrite until mid-November, specifically on the 13th. At that point, I found myself about three weeks behind schedule, and with an upcoming break on the horizon, I had to make a decision.

1. Option A: Attempt to finish the original vision I had for the game without incorporating the art aspect. However, I wasn't exactly sure how to fully implement features such as the currency system or the stat upgrading feature of the game. It would be a shot in the dark to see if I could reach that level.

2. Option B: Polish the game further, introduce some art features, and add a couple of new features that aren't as ambitious as the story mode aspect. This includes creating an A.I bot, incorporating power-ups into the game, and possibly changing the fonts.

I initially started working on Option A and even progressed to the point of creating the main screen for the story mode single-player aspect, where the player could view the table/standings with the win/lose/draw columns. However, right before the break, I switched to Option B because I wasn't sure how to implement the desired features. Considering the remaining time left, I decided to polish the game further and make it look like a completed product, rather than introducing a bunch of new features that might be 100% buggy and not as polished.

Below is what my new timeline looked like with all the adjustments and with the new MVP that I sought to achieve. As you can see, I finished refactoring my code on the 13th of November and decided to focus on just the A.I Bot, adding power-ups/refactoring that, and also creating/finding some art for my game. I also crossed out some of the features that I decided not to pursue, which would be considered a Stretch Goal if I had any extra time after completing my new initial goals.

[illegible]

4.3 Art + Final Adjustments

After adding the new features I wanted with the limited time left, the final step was to create some art. I utilized a website called PixilArt.com to craft the player's head for my game, as it was straightforward and easy to use. However, for the foot and goal art, I found suitable images online. Despite attempting to recreate these art pieces, I couldn't achieve the desired look, prompting me to search for them on the internet. Below are the links and artists' names:

1. [Goal Sprite - By Sinthea](#)
2. [Shoes Sprite - By Patinya Angsurat](#)



Figure 4.1: Control Page of my game with the Player Heads that I created

For the final adjustments to my game, I spent time experimenting with various values, such as how fast a player should move and the magnitude of a power-up boost. Additionally, I explored different fonts to determine which one complemented the vibe of my game better, as the default font in Unity didn't quite align with the aesthetic I was aiming for.

Testing

Testing in my game was quite straightforward. In the Unity Editor, there's a convenient play button that allows me to simulate exactly what the players will experience once I fully build my game. This meant that I didn't have to write any specific unit tests since each time I played the game if there was an error with my code, the editor would notify me that it couldn't compile. Additionally, even if I could run the game, I would use lots of 'Debug.Log' statements (similar to print statements) to check my console/terminal and ensure that actions were happening as intended.

With each new feature, I added to the game, I could simply hit that play button and experiment to catch any bugs or adjust values to enhance gameplay fluidity and balance.

While I must admit I didn't adhere to a set schedule or conduct daily checkups for testing, I made it a point to test my game every time I wrote new code. This practice ensured that nothing was broken, and I could quickly address any unusual errors that cropped up during development. I didn't encounter any significant struggles in testing my game.

Project Installation Guide

To get access to the game, there are only a few simple steps to follow, **however, the game is only available for Windows/PC!!!** If you are using a Mac or Linux I am sorry but this won't work on those Operating Systems.

1. **Clone the Repository** - Open the command prompt and run the following command to clone the repository on your machine. Give it a few seconds or couple of minutes for it to load on your machine. `'git clone https://github.com/jf2024/2D-Soccer.git'`
2. **Navigate to the Code Directory** - Next, after the cloning is done, go to the directory or wherever you cloned it too. `'cd path/to/2D-Soccer'`
3. **Navigate to 2D-Soccer Folder** - Once you are in the project folder, there should be a folder called 2D-Soccer that you will need to enter next. Sounds confusing since the project name is 2D-Soccer and inside of it there is another folder called 2D-Soccer but don't worry, that's how it's supposed to be, enter into that folder.
4. **Exe File and Play Game** - Once you are inside the 2D-Soccer folder, there should be .exe file called 2D-Soccer with a soccer ball logo/icon. Once you locate that icon, double click that and you should be good to go to play 2D-Soccer!

Passing the Baton

7.1 On-boarding

To onboard someone onto this project, it would be beneficial for them to have some experience with the Unity Editor and C#. Without this experience, it might be a bit challenging, but if the person is willing to invest time in learning the basics and gaining practice with the Unity Editor, that would be a significant help.

Specific examples of what someone should be familiar with include understanding how prefabs work, which serve as blueprints to create multiple objects of the same type, akin to an interface. Other essential concepts include Transforms and Components, which relate to the positions, appearance, and behavior of objects in the game space.

Delving into more technical details, a crucial aspect to understand is the Game Manager file. This file handles many of the main mechanics of the game, such as scorekeeping, timer management, and signaling the start and end of the game. It also enables players to pause the game, either returning to the main menu or resetting the current gameplay. As the hub of how everything comes together, having a good understanding of the Game Manager is vital for expanding on this project.

Once someone comprehends the Game Manager file, other aspects, such as code for player inputs, goal registration, and scoring, become more straightforward. These components don't have as much code complexity as the Game Manager, so someone coming onto the project shouldn't be overly stressed with these pieces.

In summary, having a basic understanding of Unity is crucial for onboarding onto this project. While experience with object-oriented programming and functional programming is nice to have, it's not completely required. Some programming experience, however, is necessary.

7.2 What's left?

The Minimum Viable Product (MVP) stands as a well-polished and error-free soccer game, providing players with a complete and enjoyable experience. However, for those wishing to carry forward and expand upon what I currently have, they could carry out some of my envisioned features outlined in the Stretch Goals to enhance the game's depth and engagement.

One notable avenue is the implementation of a Story Mode, something we talked about earlier in this document which transforms regular soccer matches into a comprehensive season. This narrative-driven expansion could include tournaments, leagues, and championships, providing players with a rich storyline, challenges, and objectives. Introducing a virtual currency system within this mode, earned through in-game achievements, adds a strategic layer. Players could utilize this currency to upgrade various aspects of their characters, fostering a sense of progression and customization. The Minimum Viable Product (MVP) stands as a well-polished and error-free soccer game, providing players with a complete and enjoyable experience. However, those looking to expand upon what I currently have could implement some of the envisioned features outlined in the Stretch Goals to enhance the game's depth and engagement.

One notable avenue is the implementation of a Story Mode, as discussed earlier in this document, transforming regular soccer matches into a comprehensive season. This narrative-driven expansion could include tournaments, leagues, and championships, providing players with a rich storyline, challenges, and objectives. Introducing a virtual currency system within this mode, earned through in-game achievements, adds a strategic layer. Players could use this currency to upgrade various aspects of their characters, fostering a sense of progression and customization.

Additionally, refining the AI bot presents an opportunity to elevate the game's challenge. Beyond the current implementation, improvements could involve an adaptive difficulty

system, ensuring a suitable level of challenge for players of varying skill levels. Enhancing the bot's decision-making processes through machine learning or advanced decision trees contributes to a more realistic and strategic opponent.

Visually, embracing soccer-themed art concepts and incorporating realistic player models, possibly resembling renowned real-life players, would elevate the game's authenticity. This includes refining stadium and environmental designs for a more immersive experience. Such enhancements collectively contribute to a more comprehensive and polished soccer gaming adventure, offering both depth and realism to players who choose to explore the expanded vision of your game.

Additionally, refining the AI bot presents an opportunity to elevate the game's challenge. Beyond the current implementation, improvements could involve an adaptive difficulty system, ensuring a suitable level of challenge for players of varying skill levels. Enhancing the bot's decision-making processes through machine learning or advanced decision trees contributes to a more realistic and strategic opponent.

Visually, embracing soccer-themed art concepts and incorporating realistic player models, possibly resembling renowned real-life players, would elevate the game's authenticity. This includes refining stadium and environmental designs for a more immersive experience. Such enhancements collectively contribute to a more comprehensive and polished soccer gaming adventure, offering both depth and realism to players who choose to explore the expanded vision of your game.

Final Thoughts

8.1 What went well?

One strength I demonstrated throughout the development of the game was consistently making progress, even if it meant taking a few steps back or if the progress wasn't as substantial as hoped. This continuous development was a notable achievement, with something accomplished at least once a week. Whether the progress was significant or minor, the key was moving in the right direction, and maintaining this momentum was a win.

For instance, at the start of the semester, I focused on improving the visuals by refining the home screen from a clunky mess to a more refined and cleaner layout, enhancing readability with different pages for players to navigate. Another significant achievement was rewriting my code, a tedious but essential task that highlighted the importance of doing things right the first time rather than relying on code that may or may not work without a full understanding of its long-term implications.

Moreover, I embarked on creating artwork for the game, a task I didn't initially anticipate. This step not only improved the aesthetics but also added an element of fun to the game, moving beyond the use of basic shapes like rectangles and circles to represent goals and players. Overall, the consistent and varied progress made throughout the development process stands out as a noteworthy accomplishment.

8.2 What went wrong?

In terms of challenges, a couple of things went awry during the development process. First and foremost was the complexity of my initial idea, particularly the ambitious goal of completing the story mode by the end of the semester. Despite my assumption that I could implement it after ensuring my foundation was correct, the reality proved to be different. After consulting with Professor Morgan and realizing the extensive work involved, especially given the need to solidify my foundation first, it became clear that the timeline was too tight.

Another source of frustration was a persistent bug in the game that eluded a solution for an extended period. While attempting to resolve this bug, I spent hours troubleshooting, and its persistence hindered overall progress. Eventually, I decided to shift focus to other aspects of the game, such as refining movement mechanics and implementing visual improvements, to maintain a sense of progress. While seeking assistance from another pair of eyes would have been beneficial, circumstances, including limited meetings with Professor Morgan in the second half of the semester, led me to tackle the problem independently.

Reflecting on my approach, I identified room for improvement in my schedule and timeline management. Although I aimed to make progress every week, the lack of specific weekly objectives could have hindered a more granular understanding of my project's pace. Breaking down the timeline into weekly goals might have provided clearer insights into the project's progress.

Furthermore, I could have enhanced the testing phase by involving more game testers. While my roommate was a valuable tester, having a wider range of reliable testers could have brought additional perspectives and identified potential oversights or opportunities for improvement in the game.

8.3 What would I have done differently?

Building on the insights from the previous section, I would refine my project management by breaking down objectives on a more detailed, weekly basis. Establishing specific weekly goals could provide a clearer and more granular view of project progress, helping to identify potential delays or areas for improvement.

In addition, I would prioritize establishing a more robust play-testing schedule, aiming to

involve testers every 2-3 weeks. This regular play-testing cycle would offer valuable insights into the game's development, allowing me to gauge progress, gather diverse feedback, and make iterative improvements. The continuous feedback loop could contribute significantly to refining and enhancing the overall gaming experience.

Furthermore, reflecting on the challenges with the persistent bug in the game, I recognize the importance of spending more time examining and seeking advice on my code earlier in the semester. If faced with a similar situation, I would consider restarting the game from scratch rather than attempting to patch together broken pieces. This proactive approach could save time and potentially result in a more streamlined development process.